Automated Point Cloud Correspondence Detection for Underwater Mapping Using AUVs

Marcus Hammond*, Ashley Clark*†, Aditya Mahajan*, Sumant Sharma*, and Stephen Rock*‡
*Dept. of Aeronautics & Astronautics, Stanford University, Stanford, California 94305
†NASA Ames Research Center, Moffett Field, Mountain View, California 94035
‡Monterey Bay Aquarium Research Institute, Moss Landing, California 95039
Email: {mmh13, aaclark, mahajan1, sharmas, rock}@stanford.edu

***Abstract*—An algorithm for automating correspondence detection between point clouds composed of multibeam sonar data is presented. This allows accurate initialization for point cloud alignment techniques even in cases where accurate inertial navigation is not available, such as iceberg profiling or vehicles with low-grade inertial navigation systems. Techniques from computer vision literature are used to extract, label, and match keypoints between "pseudo-images" generated from these point clouds. Image matches are refined using RANSAC and information about the vehicle trajectory. The resulting correspondences can be used to initialize an iterative closest point (ICP) registration algorithm to estimate accumulated navigation error and aid in the creation of accurate, self-consistent maps.**

The results presented use multibeam sonar data obtained from multiple overlapping passes of an underwater canyon in Monterey Bay, California. Using strict matching criteria, the method detects 23 between-swath correspondence events in a set of 155 pseudo-images with zero false positives. Using less conservative matching criteria doubles the number of matches but introduces several false positive matches as well. Heuristics based on known vehicle trajectory information are used to eliminate these.

I. Introduction

The work presented in this paper develops an algorithm for data correspondence detection within point clouds created from sonar measurements, and applies it to field data. This correspondence information can be used to correlate measurements and estimate navigation drift as part of a loop closure process – an important step for generating accurate terrain maps. If the vehicle has relatively precise inertial navigation, the need to detect correspondences can be avoided. However, when navigational drift becomes very large, a robust means for detecting loop closure events and solving the correspondence problem becomes necessary.

A number of methods have been developed to solve the loop closure problem. Some of these involve correlating a set of measurements with another set [1], either in the form of a point cloud or a 2.5 dimensional Digital Elevation Map (DEM). However, searching for correlation peaks in such data can be computationally expensive. Iterative Closest Point (ICP) methods [2] are an efficient set of algorithms capable of high-accuracy point cloud alignment. Given a reasonably good initial alignment estimate, ICP algorithms do not require a priori correspondence knowledge, but can become trapped in local minima if initialized too far from the truth [3]. For robotic applications with large sensor drift, and where GPS or other navigational infrastructure is not available, it can be difficult to provide ICP registration methods with an initial condition that is certain to converge to an accurate result. Previous work has attempted to address this ICP initialization problem by extracting distinctive keypoints from the 3D data [4], [5] or other histogram-based approaches to describing point cloud structure [6], [7]. These perform well for recognizing man-made objects, with distinctive features, in cluttered environments. Initial attempts by the authors to apply these techniques to underwater natural terrain did not produce robust matching, though further work to adapt them may yield improved results.

The method presented here seeks to solve the loop closure problem for underwater natural terrain by providing an accurate initialization for ICP methods. It does not rely on having an accurate position estimate to establish correspondence. Instead, it identifies similarities between the measurements themselves to determine that the terrain has been observed before. Borrowing from the computer vision literature, image feature extraction, description, and matching techniques normally performed on 2-D images are applied to 3-D point clouds constructed from multibeam sonar measurements. Recent work by Leines [8] applied image feature matching to point clouds of lidar measurements. The work converted raw point clouds of a mixture of urban and natural terrain to 2.5-D DEMs. This paper uses a similar approach to find correspondences in data sets devoid of artificial landmarks, and lacking a well-defined reference plane – a phenomenon that occurs when mapping environments such as an iceberg or undersea canyon.

The initial motivation for this work is a NASA ASTEP-funded mission to map free-drifting icebergs as part of a larger goal of exploring and searching for life in extreme environments. The icebergs' motion introduces large apparent odometry errors, which complicate the process of loop closure. In addition to this specific application, the method developed herein can also be applied to any mapping task where the vehicle has low-quality inertial navigation, whether due to DVL (Doppler Velocity Logger) signal dropout, low-quality or corrupted sensors, or inertial drift accumulated over long-duration missions. Section II describes the steps of the algorithm, and Section III shows experimental results. The dataset consists of multibeam sonar readings of an underwater canyon wall in Monterey Bay, California. The measurements were collected on multiple passes

978-0-933957-43-5 ©2015 MTS

around the canyon. Correspondence detection is first applied to overlapping point cloud submaps for validation, and then used to match point cloud submaps between successive passes around the canyon.

II. Technical Approach

A. Overview

The goal of this work is to solve the data correspondence problem for point clouds composed of multibeam sonar returns, without the need for accurate odometry. These correspondences can then be used to initialize registration algorithms that might otherwise fail due to local minima, enabling the creation of underwater terrain maps even when accurate navigation is not available. At a high level, the method, outlined in Figure 1, detects recognizable "landmarks" in sonar data, in order to identify when the vehicle revisits previously-traversed terrain. The procedure consists of four steps:

Image Generation: To begin, the point cloud is divided into overlapping submaps. Each individual submap is converted into a DEM image by projecting the points onto an average normal plane. Gaps within the data are filled using Gaussian smoothing, and image contrast is enhanced.

Feature Extraction: SIFT features are extracted from the DEM image using the method described by Lowe in [9].

Image Matching: SIFT features are compared between images, and Random Sample Consensus (RANSAC) with a homographic projection model is used to enforce geometric consistency to avoid false or ambiguous matches.

False Match Rejection: Application-specific knowledge can be leveraged to serve as an additional method to eliminate false matches. In some cases, in low-information natural terrain, even with the feature-level outlier rejection methods described above, the algorithm can generate false matches. For the application considered in this paper, mapping of free-drifting iceberg keels or underwater canyons, an additional outlier rejection heuristic can be performed, based on the assumption that matching image pairs should appear consecutively. This follows directly from the robot's trajectory through the cyclic environment. A Hough transform [10] is used to find the predominant match sequence, and flag all other matches as outliers. This allows the matching threshold to be set less conservatively, producing more matches without introducing incorrect information into the mapping process.

These steps are described in more detail in Sections II-B through II-E.

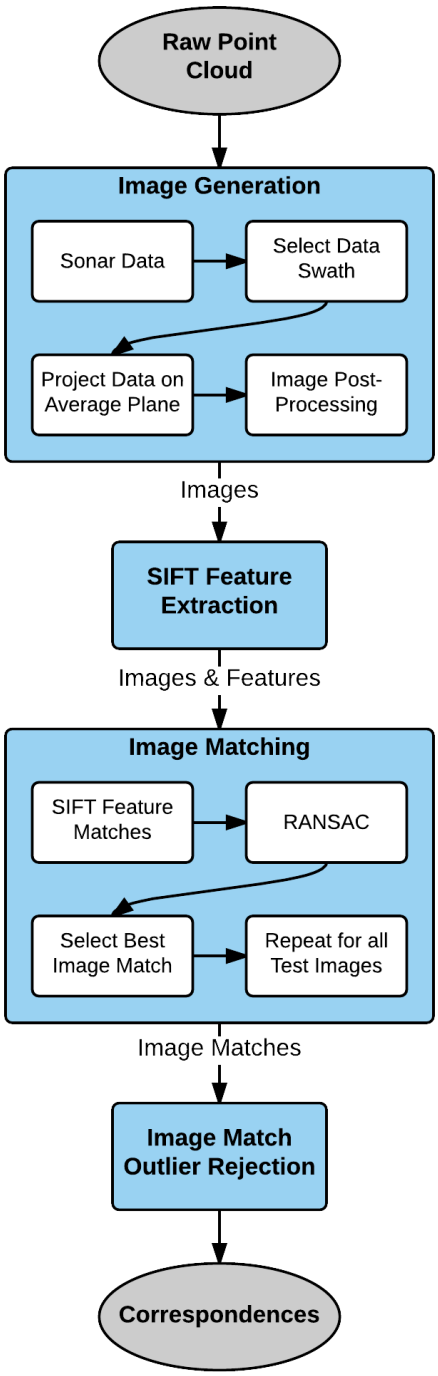

Fig. 1. Correspondences between point clouds are determined in four primary stages, some of which can be decomposed into smaller processes.

B. Image Generation

To leverage image processing tools for point cloud correspondence detection, the point cloud must first be converted into an image. The resulting grayscale images can be thought of as DEMs with pixel intensity being the "height" of the map above that location in the image.

There are several steps in creating an image. First, a reference plane and corresponding projection direction is chosen. Next, a grid is constructed on the reference plane, and the distance between a measurement and its projection onto that plane is recorded as the "intensity" at the respective location. For instance, when creating a DEM of the seafloor, the reference

plane is horizontal (x, y), and the projection direction is vertical (z). However, operating in a high-curvature environment such as icebergs or canyon walls, there may not be a single, clearly defined direction in which the projection should occur.

The method presented here calculates the average normal for a given submap and uses that as its projection direction. The reference plane is set perpendicular to this, such that all measurements lie on the positive-z side of the plane. For two partially overlapping submaps, the reference planes can have slightly different orientations.

Since image features are based on intensity gradients, and different projection planes will alter the intensity gradients within the image, the size of a submap must be chosen to minimize effects due to changes in projection direction. For small angles, the gradients are only offset by a small constant and minor scaling changes; a slight tilt in the image projection plane is comparable to a minor change in lighting angle in a standard image. Since SIFT features are known to be robust to lighting changes up to approximately 20 degrees [11], submap sizes should be designed such that the expected orientation change between consecutive images is less than this amount.

When applied to the sonar submap shown in Figure 2, data projection yields the image shown in Figure 3. Where multiple soundings project to the same pixel, the average pixel height is used in the final image. At this step, the data is also resampled on a rectangular pixel grid. As shown in the latter figure, sonar occlusions result in a number of holes or gaps in the data.

The gaps in the data are filled by a process equivalent to bilinear interpolation, in order to prevent the feature extraction algorithm from interpreting them as valid information. A mask is used to prevent features from being detected within the unobserved regions, but since the algorithm uses pixel neighborhoods to detect and describe features, the gaps must be smoothed in. The fundamental assumption behind smoothing is that the sonar returns are the result of reflecting sound energy off of an underlying physical *surface*, implying that nearby measurements will be correlated. To reflect this, this algorithm smooths the transition to the unobserved regions using an iterative procedure of eroding the full image and then restoring the known pixels to their original values. The pixels lacking data are initialized with an intensity of zero. The image is then convolved with a Gaussian kernel, acting as a low-pass filter, effectively smearing the image and filling in some of the gaps. The second step restores the pixels that began with valid data to their original values. The result is equivalent to bilinear interpolation in pixel space, but requires less bookkeeping. The result of this process can be seen in Figure 4.

The last step in creating a SIFT-ready image is Contrast-Limited Adaptive Histogram Equalization (CLAHE), shown in Figure 5. Algorithms like SIFT use gradient information in images to extract features. The CLAHE process attempts to enhance details without skewing the overall image contrast level [12].

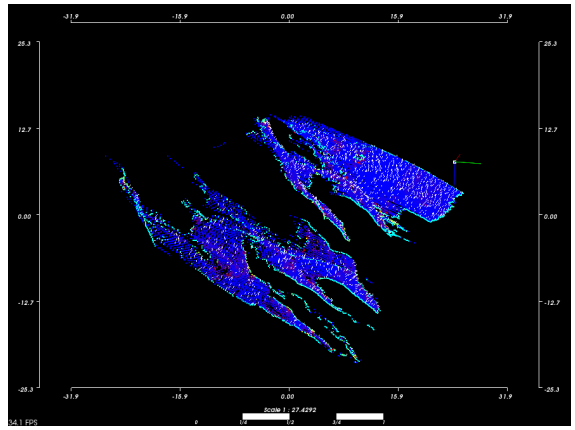

Fig. 2. A submap consists of approximately 180,000 sonar measurements. 600 scans, taken over 200 seconds were collated to generate this submap spanning 300m of terrain.

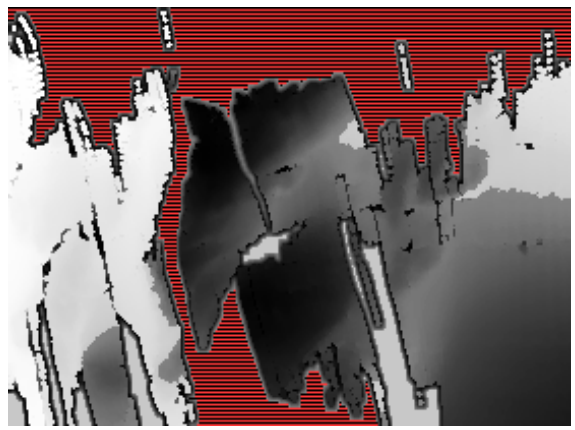

Fig. 3. The data shown in Figure 2 is converted into an image by projecting each point onto an average normal plane. Any gaps in the data are shown as red hatches.

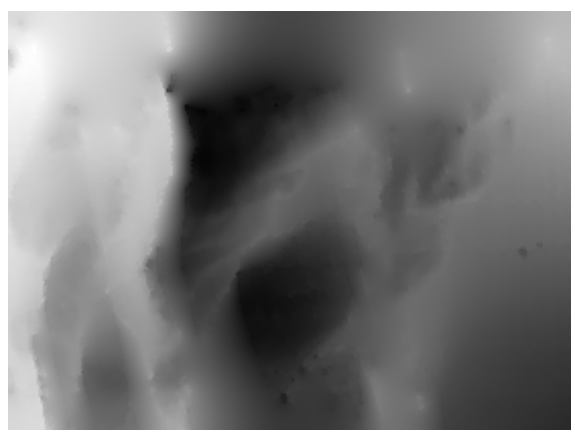

Fig. 4. After the erode-restore process, the (red hatched) gaps highlighted in Figure 3 are smoothed out. Note that the original data measurements remain unchanged.

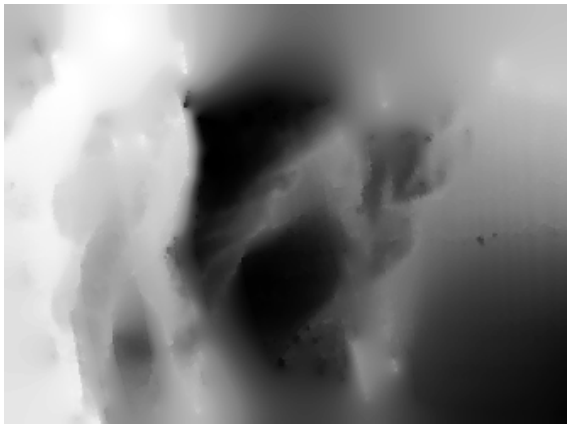

Fig. 5. Adaptive Histogram Equalization on Figure 4 boosts the contrast of the image, increasing the number of detected SIFT features.

C. Feature Extraction

After the 3D data is converted into DEM format, robust image features are extracted and used to look for loop closure events. Features located in regions with no data measurements are discarded. The work presented here employs SIFT features, though other feature classes can be used. Figure 6 shows features extracted from such an image.

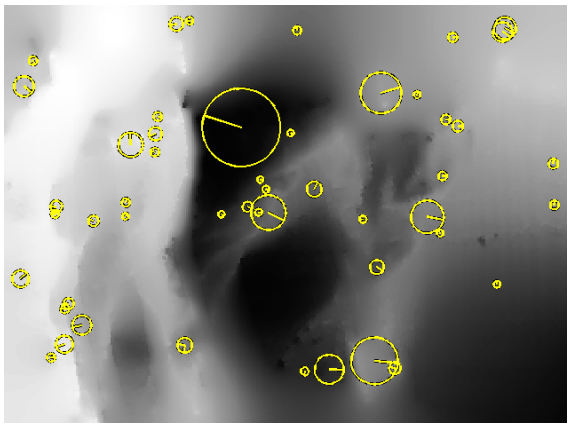

Fig. 6. A random sample of SIFT features extracted from the range image in Figure 5. Only features occurring in regions of original data are considered for matching.

D. Image Matching

To find correspondences between two datasets, each image from the second dataset is compared to each image from the first dataset, using the standard matching algorithm proposed by Lowe [9]. Two image features are considered a good match if the Euclidean distance between their descriptors is 1.5 times less than the next best feature match. Next, further spurious feature matches are eliminated using RANSAC with a homographic projection model. If RANSAC finds a valid transformation model between the two images with a sufficient number of additional inliers, the pair is labeled a mach. The required number of inliers is determined by a user-specified threshold, and constitutes a trade-off between detecting a large number of matches with moderate confidence or a small number of matches with high confidence, essentially moving the solution along a receiver operating characteristic curve. If two images from the first dataset have the same number of RANSAC inliers with a given image from the second dataset, the match with the lowest feature reprojection error is selected. Example results are shown in Figure 7.

E. Application-specific False-Match Rejection

The image matching of Section II-D leads to images being paired with their most likely counterpart from another instance during the mission. However, if false matches are a concern for the particular algorithm use case, then a strategy for false match rejection should be employed.

Application-specific knowledge can be used to perform false match detection and elimination. For the motivating application used in this paper, mapping free-drifting icebergs and underwater canyons, it is assumed that the vehicle moves at near constant velocity during data collection, and that a repeated pass of the same terrain should result in a sequence of corresponding images. If an image matches one from an earlier time in the mission, it is very likely that the next image during a pass will match the next image in the following pass. Under these assumptions, plotting the best matching image numbers from one pass against the other should yield a straight line.

In this paper, a Hough transform [10] is used to estimate the parameters of the predominant line through the image matches. Line parameters (slope and intercept) are estimated using every possible pairwise combination of datapoints and placed into discretized bins. The most populated bin is selected and these parameters represent the best linear trend of the data. Image matches that fall within a threshold distance of the line are retained, and considered inliers, whereas matches that fall outside this limit are discarded as outliers.

The final image matches output from the algorithm can be referenced back to their original point cloud data, and the resulting relative offsets can be utilized to initialize an ICP algorithm.

III. Experimental Results

The dataset used for these experiments was collected using a multibeam sonar on board an Autonomous Underwater Vehicle (AUV) in Soquel Canyon, Monterey Bay, California. The vehicle circumnavigated the canyon two and a half times, measuring ranges to the walls. The first set of experiments searches for correspondences between sequential pseudo-images with known overlap, in order to validate the matching technique. The second set then searches for correspondences between different passes along the canyon.

Each image was created using point cloud submaps of approximately 180,000 data points. The multibeam sonar collects a line-scan measurement of 300 points at a rate of 3Hz, and the

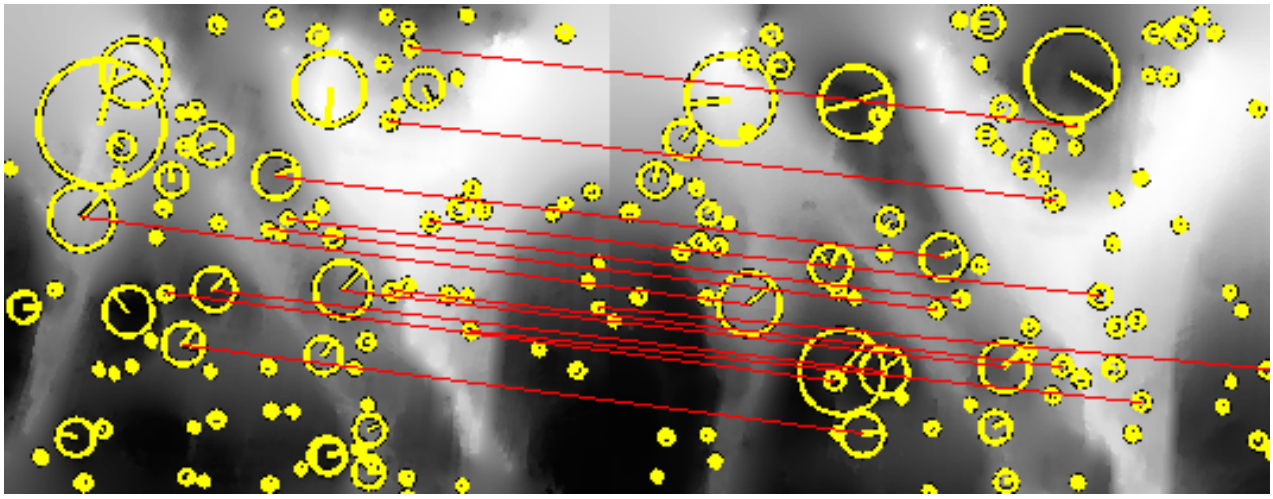

Fig. 7. SIFT feature matches between two successive passes along the Soquel Canyon wall. The features are shown as yellow circles and matching features are connected in red. Not all features are registered across both images but there are sufficient correspondences to declare an image match.

vehicle has an estimated velocity of 1.5 meters per second. The resulting images typically cover a 200 meter wide swath with a length of 300 meters (lengths are approximate due to variations in the terrain). Designed to have 75% overlap with its immediate neighbors, each image shares data points with up to three other images in either direction of travel. Overall, the entire dataset has been segmented into 406 images.

Experiments were run using two different thresholds for the number of post-RANSAC feature matches required for an image match. Each image contains between 70 and 120 SIFT features. For the higher RANSAC inlier threshold, two images are considered a match if the homography produced by RANSAC has at least 15 inliers. This yields a small number of high-quality matches with no false positives, but fails to detect some good matches in areas with fewer features. In order to detect these additional matches, a lower RANSAC inlier threshold is required. Using a RANSAC inlier threshold of 7 doubles the number of matches detected, but also detects some false positive matches. However, this can be mitigated using an application-specific outlier rejection method based on the Hough Transform.

Outlier rejection is then performed by using a Hough transform to find the predominant line through the image matches. Every pair of image matches is used to vote for line parameters. The bin with the most votes is identified, and linear regression is performed on the match pairs that contributed to this bin. Due to the 75% overlap between adjacent images, the resulting line is considered to be the center of the correct match, and all image matches within a distance of 3 from the resulting line are considered to be "good" matches.

A. Matching Overlapping Data

The technique is first validated on images with guaranteed overlap. For this test, all even-numbered images were compared to every odd-numbered image in the dataset. Given the 75% overlap between adjacent images, a match was considered correct if the algorithm identified a match between the ith even image and the $(i-1)^{\text{th}}$, i^{th}, $(i+1)^{\text{th}}$, or $(i+2)^{\text{th}}$ odd image.

Figure 8 shows results using the high RANSAC inlier threshold of 15 image feature matches. All but two of the matches correspond to a pair of overlapping submaps; the two matches that deviate from this line are in fact the same canyon location, matched between passes 1 & 3, and passes 2 & 3 respectively. Therefore, the algorithm is able to identify 132 matching images with no false positives. Lowering the RANSAC inlier threshold from 15 to 7 generates 23 additional image matches, resulting in 155 matching images with no false positives. Low RANSAC inlier threshold results are shown in Figure 9.

The images for which no correspondences are found occur in smooth, featureless terrain, with a distinct lack of easily identifiable features. These results show that for the given dataset, there is sufficient terrain texture for image matching, and that for the given image generation parameters, the variation in projection plane orientation is small enough to allow for accurate image matching.

B. Matching Multi-Pass Data

Next, the image matching algorithm is applied to longer-duration data collection passes over the same region. In this case, the 155 images of the first pass around the canyon are compared to 156 images from the second pass around the canyon. To generate truth, all images from the second pass were visually inspected and hand-labeled with the indices of all partially-overlapping images from the first pass.

Figure 10 shows results using a RANSAC inlier threshold of 15. The green shaded region denotes regions of true image overlap. The algorithm identified 23 correspondences, with no false positives. When there is a sufficiently high RANSAC inlier threshold for image matches, outlier rejection has no impact upon the accuracy of the results.

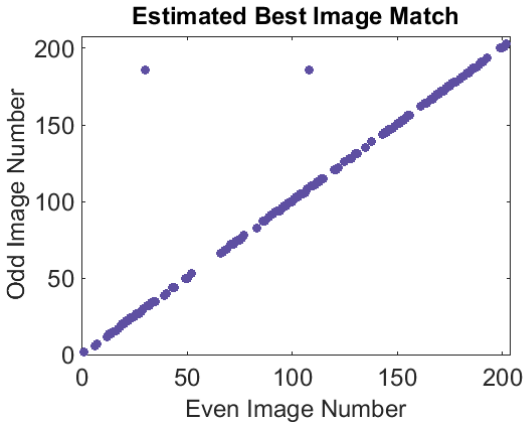

Fig. 8. Comparison of odd-numbered images with even-numbered images for a RANSAC inlier threshold of 15. This results in 132 matching images with no false positives. The two points that deviate from the predominant line are correct matches between images of the same canyon region, collected on different passes.

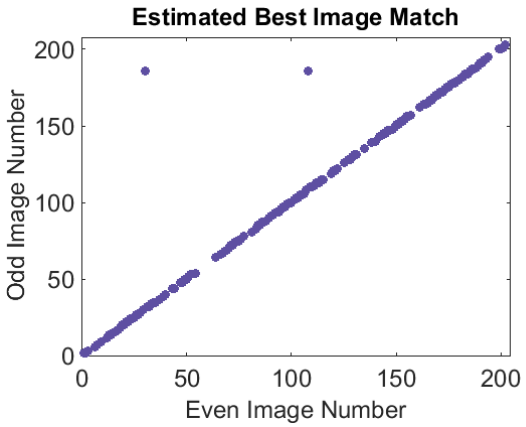

Fig. 9. Comparison of odd-numbered images with even-numbered images for a RANSAC inlier threshold of 7. This results in 155 matching images with no false positives. The two points that deviate from the predominant line are correct matches between images of the same canyon region, collected on different passes.

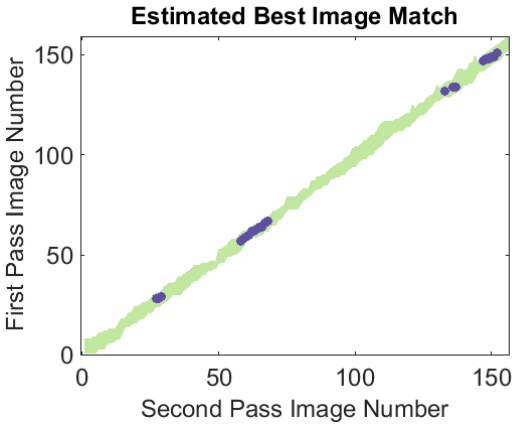

Fig. 10. Results of matching multi-pass imagery with a RANSAC inlier threshold of 15. The green shaded region denotes regions of true image overlap. Since there are no false positives, Hough transform outlier rejection has no impact upon the results.

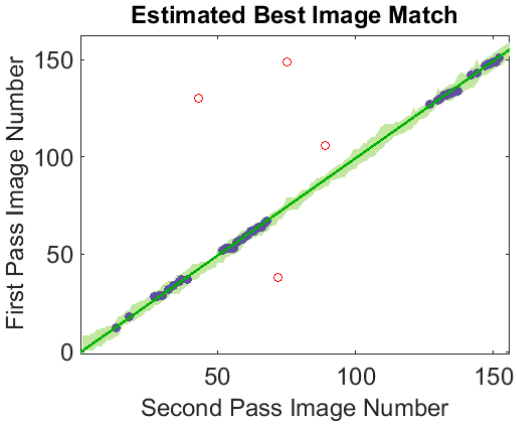

Fig. 11. Results of matching multi-pass imagery with a RANSAC inlier threshold of 7. The lower threshold allowed more good matches to be detected (solid circles), but also produced some false matches (hollow circles). These false matches were rejected using a Hough Transform, which gives a line-of-best-fit (green).

Lowering the RANSAC inlier threshold introduces additional true correspondences, as well as four false positives. Results for a RANSAC inlier threshold of 7 are shown in Figure 11. Inlier matches are plotted as solid circles, and outliers are plotted as hollow circles, with the Hough line of best fit shown in green. There are a total of 49 image matches prior to outlier rejection, 45 of which are correct. Outlier rejection correctly identifies the inaccurate image matches, retaining the 45 matches with no false positives. By lowering the RANSAC inlier threshold and applying application-specific outlier rejection, the number of detected correspondences is doubled.

IV. Conclusions

Experiments on the Soquel Canyon dataset show that the algorithm presented here is able to identify correct correspondences between sets of point clouds. The ability to detect correspondences in range data taken from natural terrain allows loop closure detection even after the accumulation of large navigational drift, which is valuable for iceberg profiling and mapping of static terrain using vehicles with low-precision inertial navigation systems.

Using strict matching criteria, the method identifies no false matches, but at the cost of rejecting a number of good matches. Using less conservative matching criteria doubles the number of matches but introduces several false positive matches as well. Heuristics based on known vehicle trajectory information are used to eliminate these.

Once image correspondences are correctly identified, the information can be used to accurately initialize an ICP algorithm for the purpose of estimating accumulated navigational drift.

Acknowledgments

This work was funded under NASA ASTEP Grant NNX11AR62G. The authors would also like to thank the

Monterey Bay Aquarium Research Institute for the data, ship time, and technical support. Particular thanks go to Brett Hobson, Hans Thomas, Rob McEwen, and Rich Henthorn.

REFERENCES

[1] A. E. Johnson and J. F. Montgomery, "Overview of terrain relative navigation approaches for precise lunar landing," in *IEEE Aerospace Conference Proceedings*, 2008.

[2] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Proc. SPIE*, vol. 1611, 1992, pp. 586–606.

[3] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, 2004, pp. 22–31. [Online]. Available: http://dl.acm.org/citation.cfm?id=1057435

[4] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Symposium on Geometry Processing*, 2005, pp. 197–206.

[5] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 5 2009, pp. 3212–3217.

[6] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 5 1999.

[7] a. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3d point clouds," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, vol. 1. IEEE, 2006, pp. 1297–1304.

[8] M. T. Leines and J. F. Raquet, "Terrain reference navigation using sift features in lidar range-based data," in *Proceedings of the 2015 International Technical Meeting of The Institute of Navigation*, 2015, pp. 239 – 250.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 11 2004.

[10] P. V. C. Hough, "Method and means for recognizing complex patterns," 1962.

[11] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A Comparative Study of SIFT and its Variants," *Measurement Science Review*, vol. 13, no. 3, Jan. 2013. [Online]. Available: http://www.degruyter.com/view/j/msr.2013.13.issue-3/msr-2013-0021/msr-2013-0021.xml

[12] S. Pizer, R. Johnston, J. Ericksen, B. Yankaskas, and K. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," in *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*. IEEE Comput. Soc. Press, 1990, pp. 337–345.